# Sphinx & Read The Docs Documentation Steps

This guide documents every step and decision made to set up Python documentation using Sphinx and Read The Docs for your project. Use this as a future reference for yourself or anyone working on this repository.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 1. Project Structure

Create an empty docs/ folder at the root of the package (not inside src/)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 2. Install Sphinx And Extensions

Open a terminal and run:

pip install sphinx sphinx\_rtd\_theme sphinx-autodoc-typehints

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 3. Initialize Sphinx

From attendance\_tool\_msp/docs:

cd attendance\_tool\_msp/docs

sphinx-quickstart

- Answer prompts for project name, author, etc.

- This creates source/ folder with conf.py, index.rst inside & Makefile, make.bat, etc. At the root of docs (not inside source)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 4. Configure Sphinx To Find Your Package

This path lets Sphinx find your code for API documentation (conf.py in other words).

Adjust the path if you move your folders—always point to the directory containing your package.

In docs/source/conf.py, add at the very top:

import os

import sys

sys.path.insert(0, os.path.abspath('../../src'))

# Points to the src code of our package!

Set extensions:

extensions = [

'sphinx.ext.autodoc',

'sphinx\_autodoc\_typehints',

'sphinx.ext.napoleon',

]

Set theme (Override the current one in file):

html\_theme = "sphinx\_rtd\_theme"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 5. Generate .rst Files For Your Modules

From attendance\_tool\_msp/docs:

sphinx-apidoc -o source ../src/attendance\_tool\_msp

- This creates attendance\_tool\_msp.rst, attendance\_tool\_msp.gui.rst, etc.

- MAKE SURE modules.rst EXISTS & Place your modules as text inside if not already

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 6. Simplify .rst Files For Clean API Documentation (To Avoid Lots Of Nesting & Complicated Docs)

.rst files are files formatted using reStructuredText (RST), a plaintext markup language primarily used for technical documentation, especially within the Python programming language community.

Edit attendance\_tool\_msp.rst and attendance\_tool\_msp.gui.rst to remove deep nesting and submodules. Keep only:

.. automodule:: attendance\_tool\_msp

:members:

:undoc-members:

:show-inheritance:

.. automodule:: attendance\_tool\_msp.gui

:members:

:undoc-members:

:show-inheritance:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 7. Edit index.rst For Project Description And Navigation

At the top of docs/source/index.rst, add your project description (Works Just Like HTML!):

An easy-to-use Python package for automated attendance processing, validation, and professional report generation. Built mainly for Microsoft Student Partners Tech Club (MSP), but works for any technical session or event at Misr International University (MIU).

Below, add your 2 modules (IMPORTANT FOR HTML FILES GENERATION):

.. toctree::

:maxdepth: 2

:caption: API Reference

attendance\_tool\_msp

attendance\_tool\_msp.gui

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 8. Build HTML Documentation

From attendance\_tool\_msp/docs:

make html

- On Windows, use: make.bat html OR .\make html

- View docs in docs/build/html/index.html (Open with live server to check out!)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 9. .gitignore Setup

Add these to .gitignore to avoid pushing build and cache files:

docs/build/

docs/source/\_build/

\_\_pycache\_\_/

\*.pyc

\*.pyo

\*.pyd

\*.egg-info/

dist/

build/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 10. Push To GitHub

- Commit and push all source code, documentation source files, Makefile, and make.bat.

- Do NOT push docs/build/ or generated HTML files. (Auto-generated by readthedocs)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 11. Set Up Read The Docs

- Go to https://readthedocs.org/

- Import your GitHub repository

- RTD will auto-detect your Sphinx docs and build them

- Add RTD badge/link to your README for easy access

Import your GitHub repository on Read The Docs.

Add a .readthedocs.yaml file at the root of your repository.

This file tells RTD where your Sphinx config is and how to install dependencies.

Create a requirements.txt inside your docs folder (where your Sphinx config lives).

List all needed Sphinx extensions in requirements.txt (like sphinx, sphinx\_autodoc\_typehints, sphinx\_rtd\_theme).

In .readthedocs.yaml, use the python.install section to point to your requirements.txt.

Make sure the sphinx.configuration key matches the actual path to your conf.py.

By doing this, RTD will use your config and install all the required Python packages, so your build will work. (Important requirements.txt or build will always fail)

If you get build errors (check out raw log) about missing modules or config validation, check the paths in .readthedocs.yaml and make sure all extensions are listed in requirements.txt.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Notes:

- Sphinx only needs your source files, not generated HTML.

- You can always regenerate .rst files with sphinx-apidoc if your code changes.

- Keep docs/ outside src/ for clarity and best practices.

- Makefile and make.bat are useful for local builds—keep them in version control.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This roadmap covers every step and decision for future reference. Edit as needed for your workflow!